IN THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application.

1-75. (cancelled)

76. (previously presented) The method of claim 81, wherein in step (a)

a flip-flop is marked, if any of its latch interface constraints belong to the said unsatisfiable core, and

a flip-flop is marked, if its initial state value constraint belongs to the said unsatisfiable core.

77. (previously presented) The method of claim 81, wherein in step (a)

a flip-flop is marked, if any of its latch interface constraints belong to the said unsatisfiable core, and

a flip-flop is unmarked if only its initial state value constraint belongs to the said unsatisfiable core.

78. (currently amended) The method of claim 81, wherein during the bounded model checking

a modified initial value constraint is used for flip-flop, with an initial value constraint m being replaced by ~~(m+y)(m+!(not(y))~~(m+y)(m+y'), where y is a fresh variable not used in rest of the satisfiability-based check ~~formula~~.

79. (currently amended) The method of claim 81, wherein during the bounded model checking

a modified constraint is used for an external constraint node, with an environmental constraint (m) being replaced by ~~(m+y)(m+!not(y))~~(m+y)(m+y'), where y is fresh variable not used in rest of the satisfiability-based check ~~formula~~.

80. (cancelled)

81. (currently amended) A computer implemented method for verification of a given correctness property on a sequential circuit design by using a satisfiability-based check for bounded model checking, comprising:

(a) if the given correctness property is proved correct at a depth *k*, marking only flip-flops and external constraint nodes in the circuit design ~~are marked~~, based on whether their constraints appear in an unsatisfiable core generated from ~~the~~a proof of unsatisfiability by ~~the~~a satisfiability solver; otherwise, terminating verification,

(b) deriving an abstract model consisting of combinational fanin cones of the marked flip-flops and external constraint nodes,

(c) checking the given correctness property on the derived abstract model, and

- if the derived abstract model is proved correct, terminating the verification and deeming the circuit design to be correct.
- otherwise, increasing the depth of unrolling *k* and repeating steps (a)-(c) until either the given correctness property is violated at some depth or the circuit design is determined to be correct.

82. (currently amended) A computer implemented method for verification of a given correctness property on a sequential circuit design, comprising the steps:

(a) if the property is violated at depth k using a satisfiability-based bounded model checking, ~~reporting a counterexample is reported if it is real and~~ terminating the verification,

(b) if the property is proved correct at depth k using a satisfiability-based bounded model checking, deriving an abstract model A_n from ~~the~~a proof of unsatisfiability generated by the satisfiability solver,

(c) increasing the depth of unrolling k in bounded model checking,

(d) repeating steps (a – c), constituting an *inner loop* comprising the bounded model checking iterations, until either there in no change in the size of the derived abstract model as the depth of unrolling k is increased, or some limit $kmax$ on k is reached,

(e) repeating steps (a – d) by using bounded model checking on the derived abstract model A_n to derive a new abstract model A_{n+1}, these steps constituting an outer *loop* comprising the abstraction iterations, until either there is no change in the size of the derived abstract model or some limit on number of abstraction iterations is reached,

(f) checking the given correctness property on the derived abstract model, and if the derived abstract model is proved correct, deeming the circuit design to be correct; otherwise, deeming the verification to be inconclusive.

83. (previously presented) The method of Claim 81, where the size of the derived abstract model corresponds to the number of flip-flops in the derived abstract model.

84. (cancelled)